# Lógicas Temporales Ejecutables

I. P. de Guzmán Molina, P. Cordero, M. Enciso, C. Rossi, J. F. Moncada

Grupo de Investigación de Matemática Aplicada en Computación (GIMAC)
E.T.S.I. Informática
Universidad de Málaga
29071. Málaga (España)

## Resumen

El presente documento recoge las principales líneas de trabajo desarrolladas en el área de *modelización y razonamiento temporal* por el grupo de investigación de Matemática Aplicada en Computación de la Universidad de Málaga.

## 1. Introducción

El trabajo de nuestro grupo en el campo de la modelizacion y razonamiento temporal, desarrollado en los últimos 10 años, está centrado en el uso de la Lógica Temporal en Computación. A nuestro entender, dos son las asignaturas pendientes de esta herramienta formal en su lucha por ocupar el puesto que les corresponde en las Ciencias de la Computación:

— Para su uso como *herramienta de especificación formal* de sistemas, es necesario disponer de un eficaz tandem sintáctico-semántico; es decir:

  — *Conectivas apropiadas*, que permitan una especificación natural de propiedades reclamadas por las aplicaciones.

  — *Semántica natural*, fácil de usar, que refleje adecuadamente la naturaleza del tiempo sobre la que se está trabajando y que conserve una intuición del significado de las conectivas.

— Para su uso como *herramienta de control*, es necesario disponer de *mecanismos de deducción automática* eficientes. Dichos mecanismos serán más útiles si se enmarcan en una metodología general, es decir, siempre es deseable un demostrador automático de teoremas antes que un método de análisis de modelos (model checking); asímismo, será preferible un demostrador no clausal, antes que uno que exiga el previo tratamiento de las fórmulas para convertirlas a forma clausal.

El contenido del trabajo se ordenará siguiendo el esquema previo; es decir, comentaremos los resultados principales aportados por nuestro grupo en cada uno de los aspectos que acabamos de mencionar. Enmarcaremos cada contribución en el campo del razonamiento temporal, citando los puntos de partida en cada caso.

## 2. Nuevas Lógicas Temporales

El trabajo de investigación desarrollado en este tema tiene como referencia más próxima los trabajos del grupo de investigación que lidera el profesor Dov M. Gabbay en el Imperial College de Londres. La propuesta de dicho grupo incluida en [Gab89], presenta a la lógica temporal bajo un nuevo paradigma llamado *Pasado Declarativo, Futuro Imperativo.* Dicho paradigma propone la relectura de las fórmulas de la lógica temporal (donde hay conectivas de pasado, de presente y de futuro) en el modo:

*Si $A$ sucedió en el pasado, entonces ejecutar $B$.*

En dicho trabajo se demuestra que dicho paradigma es aplicable a las lógicas totalmente expresivas (como la lógica US de Hans Kamp), que poseen la llamada *Propiedad de Separación*, que asegura que cualquier fórmula $A$ es traducible (equivalente) a otra fórmula construida como combinación booleana de tres fórmulas ($A_1, A_2$ y $A_3$) que, respectivamente, sólo tienen información de pasado, presente y futuro.

Nuestro trabajo se ha centrado en la búsqueda de nuevas lógicas temporales totalmente expresivas (para poder ser usadas bajo el paradigma de Dov Gabbay) cuyas conectivas representen propiedades de interés en computación y en el desarrollo de una nueva semántica que evite el uso de la Lógica de Primer Orden como teoría de modelos.

## 2.1. Nuevas Conectivas

En este apartado, se ha desarrollado una familia de lógicas temporales determinadas por la elección de los siguientes parámetros dependiendo del problema a tratar:

— El flujo de tiempo puede ser continuo o discreto.

— Puede ser una lógica de puntos y/o intervalos.

— Puede tener una visión relativa o absoluta del tiempo.

En esta discusión, por razones de espacio, nos limitaremos a la consideración de flujos de tiempos lineales. Asímismo, no incluimos la posibilidad de contemplar lógicas de coordenadas (Modal $\times$ Temporal).

El punto de partida para la elección de las conectivas es la consideración de que gran parte de las aplciaciones que requieren el uso de lógica temporal están ligadas a la existencia (o necesidad de ejecución) de dos (o más) sucesos en el tiempo, y en comprobar (o establecer) su ordenación temporal.

### 2.1.1. Lógica LN

La lógica Last-Next (LN) [BG92] es una lógica temporal sobre tiempo lineal, discreto e infinito. Dicha lógica es equivalente a la lógica $US$ y, por tanto, tiene total potencia expresiva.

En el trabajo [BG92] se presentan dos conjuntos equivalentes de conectivas para LN, basados en la comparación temporal de la aparición de dos sucesos. Esta aparición puede ser *Fuerte* o *Débil* (según se exija o no su aparición).

Así, tenemos las siguiente conectivas temporales de precedencia fuerte de futuro y pasado, respectivamente:

$A \preceq B$: $A$ sucederá en el futuro y la primera vez que suceda $A$ será anterior o simultánea con la primera aparición de $B$.

$A \succeq B$: $A$ sucedió en el pasado y la última vez que sucedió $A$ fue posterior o simultánea con la última aparición de $B$.

Y sus equivalentes conectivas de precendencia débiles:

$A \sqsubseteq B$: Si $A$ sucede en el futuro, entonces la primera vez que suceda $A$ será anterior o simultánea con la primera aparición de $B$.

$A \sqsupseteq B$: Si $A$ sucedió en el pasado, entonces la última vez que sucedió $A$ fue posterior o simultánea con la última aparición de $B$.

Como sistemas de conectivas derivadas de estos primitivos obtenemos los siguientes sistemas :

— Conectivas de simultaneidad (para especificar la aparición simultánea de sucesos en el tiempo) fuertes $\{\approx^+, \approx^-\}$ y débiles $\{=^+, =^-\}$.

— Conectivas de precendencia estricta (para especificar la aparición de sucesos separados en el tiempo) fuertes $\{\prec, \succ\}$ y débiles $\{\sqsubset, \sqsupset\}$.

## 2.2. La Lógica $\mathbb{R}$LN

La elección de una lógica basada en un flujo de tiempo discreto o continuo depende del problema particular que se trata de resolver. Aunque para una gran variedad de problemas son suficientes las lógicas sobre tiempo discreto [AM87], [Gab89], las lógicas sobre tiempo continuo no pueden ser obviadas, como afirman destacados autores [BPK86], [Wol93], [Gab92].

Nuestro grupo ha desarrollado una extensión natural de la lógica LN para tiempo continuo, denominada $\mathbb{R}$LN [EdG93a], que mantiene las propiedades de LN.

En particular, $\mathbb{R}$LN es adecuada para la especificación de los llamados *sistemas reactivos* (sistemas que interactúan continuamente con su entorno). [1] En los sistemas reactivos se da una mezcla no trivial de componentes discretas (programadas) y continuas (propias del entorno). Por otra parte, algunos sistemas reactivos tienen componentes asíncronas que provocan cambios de estado que no se ajustan a los ticks de un reloj global; por lo tanto, es necesario un conjunto denso y continuo para modelizar el flujo del tiempo.

La anterior diferenciación entre hechos del programa y hechos del entorno marca en cierta medida la construcción de las numerosas conectivas definibles en $\mathbb{R}$LN. Éstas se pueden agrupar según su aridad: [2]

— Conectivas monarias:

  — $Ac^+(A)$: si $A$ se da en el futuro, $A$ será cierta en instantes arbitrariamente cercanos al instante actual

  — $Al^+(A)$: en el futuro hay un primer instante accesible en el que $A$ será cierto

— Conectivas binarias. Definimos tres sistemas distintos de conectivas de este tipo (como extensiones naturales de las conectivas de LN):

  — $A \preccurlyeq B$: en el futuro hay un primer instante en el que $A$ es verdadero, y este será, anterior o simultáneo a cualquier aparición de $B$

  — $A \sqsubseteq B$: si en el futuro ocurre $B$, entonces habrá una aparición de $A$ anterior o simultánea a cualquier aparición de $B$

  — $A \Subset B$: si en el futuro ocurre $B$, entonces habrá apariciones de $A$ antes de cualquier aparición de $B$, o bien no es posible determinar si $A$ precede a $B$ o si $B$ precede a $A$

A partir de estas conectivas se construyen tres sistemas que se demuestran totalmente expresivos:

— $\mathcal{C}_T^1 = \{\preccurlyeq, \succcurlyeq, Ac^+,\ Ac^-\}$

— $\mathcal{C}_T^2 = \{\sqsubseteq, \sqsupseteq\}$

— $\mathcal{C}_T^3 = \{Al^+, Al^-, \Subset, \Supset\}$

[1] Por ejemplo, casi todos los sistemas concurrentes, distribuidos o embebidos son reactivos.

[2] Indicamos únicamente las de futuro.

## 2.3. LNint: una lógica de puntos e intervalos

En las aplicaciones, a la hora de elegir una lógica temporal, se encuentran en la literatura posturas encontradas: por una parte, se tiene la clásica discusión entre lógicas de puntos y lógicas de intervalos; por otra parte, parece necesario escoger entre un modelo de tiempo absoluto (como en las lógicas reificadas o de argumentos temporales) o relativo (como en las lógicas modales).

En nuestra opinión, no se trata de una elección subjetiva, sino que cada aplicación reclamará una o varias de las características citadas y, en lo referente al primer aspecto, es difícil pensar en aplicaciones que tan solo requieran puntos o solo intervalos. Este planteamiento fue nuestro punto de partida para diseñar una lógica [3] que combinase todas estas características.

LNint [dGR95] es una lógica temporal sobre tiempo discreto para manejar conjuntamente puntos e intervalos. A nivel sintáctico, consideramos dos componentes, una para especificar afirmaciones sobre puntos, fechas y puntos pertenecientes a intervalos, y otra para especificar los *eventos*. [4] Ambas componentes iniciales, separadas en principio, son extendidas posteriormente para alcanzar finalmente una cohabitación semántica perfecta. Esta cohabitación se basa en la siguiente idea: en LNint hablamos de puntos o intervalos, pero siempre situados en un instante (de evaluación), el *instante actual* o presente. Más adelante, evitamos el ambiguo concepto de *intervalo actual* impuesto por las lógicas modales, que no es muy intuitivo ni tratable computacionalmente.

En la construcción del lenguaje de LNint se comienza distinguiendo entre tres tipos de enunciados atómicos:

— Aquellos cuya ejecución tiene lugar en un instante.

— Aquellos que nombran un instante.

[3] Dentro de la familia de lógicas con semántica topológica.

[4] Empleamos el concepto de evento en un sentido similar a Allen, es decir, como expresiones sobre intervalos que no son ciertas en los subintervalos ni, más específicamente, en los puntos del intervalo en el que se afirma la expresión.

— Aquellos que refieren eventos.

Nuestro objetivo de un lenguaje para hablar simultáneamente de puntos e intervalos conlleva la definición previa de diversos sublenguajes:

### 2.3.1. El "lenguaje de eventos" $\mathcal{L}_{int}$

En este sublenguaje contemplamos que los eventos pueden ser tratados de dos formas diferentes:

- *de manera relativa*, expresando la relación entre eventos. Allen [All83] considera que hay doce posibles relaciones temporales entre dos eventos y, posteriormente, Shoham [Sho88] muestra que basta con seis de ellas, (las seis restantes pueden ser definidas a partir de las primeras), recogidas en las seis conectivas básicas de nuestro alfabeto: $\langle ab^+\rangle$, $\langle ab^-\rangle$, $\langle beg\rangle$, $\langle \widehat{beg}\rangle$, $\langle end\rangle$, $\langle \widehat{end}\rangle$.

  Estas conectivas temporales de eventos pueden ser descritas informalmente como sigue:

  — $\langle ab^+\rangle\mathcal{A}$: $\mathcal{A}$ es verdadera en un intervalo contiguo por la derecha (futuro) al actual.

  — $\langle beg\rangle\mathcal{A}$: $\mathcal{A}$ es verdadera en un subintervalo inicial estricto del actual.

  — $\langle end\rangle\mathcal{A}$: $\mathcal{A}$ es verdadera en un subintervalo final estricto del actual.

  — $\langle ab^-\rangle\mathcal{A}$: $\mathcal{A}$ es verdadera en un intervalo contiguo por la izquierda (pasado) al actual.

  — $\langle \widehat{beg}\rangle\mathcal{A}$: $\mathcal{A}$ es verdadera en un intervalo del cual el actual es subintervalo inicial estricto.

  — $\langle \widehat{end}\rangle\mathcal{A}$: $\mathcal{A}$ es verdadera en un intervalo del cual el actual es subintervalo final estricto.

- *de manera absoluta*, es decir, ligando los eventos con intervalos de extremos (fechas) conocidas.

### 2.3.2. El "lenguaje de puntos" $\mathcal{L}_p$.

Con este sublenguaje pretendemos formalizar afirmaciones que hablen sobre puntos, afirmaciones sobre intervalos heredadas por los puntos que forman el intervalo, y afirmaciones sobre fechas.

$\mathcal{L}_p$ se puede entender como el lenguaje de la lógica LN, extendido para poder hablar de fechas.

En $\mathcal{L}_p$ introducimos, como definida, una conectiva absoluta que asocia una afirmación puntual y una fecha:

$$A\ at\ \underline{m} = (A \wedge \underline{m}) \vee A\ atnext\ \underline{m} \vee A\ atlast\ \underline{m}$$

$at\underline{m}$ nos permite saber si una fórmula es verdadera en un instante determinado, independientemente del instante desde el que hablamos, es decir, logramos relativizar el instante actual.[5]

### 2.3.3. El "lenguaje de puntos y eventos" $\widehat{\mathcal{L}_p}$

Definidos $\mathcal{L}_{int}$ y $\mathcal{L}_p$, extendemos éste último con el siguiente objetivo: obviar el uso intrínseco del lenguaje de intervalos, manipulando éstos mediante puntos. Entendemos que, si bien es cierto que los eventos, por su propia naturaleza, no tienen sentido sobre los puntos del intervalo sobre el que se afirma, no es menos cierto que todo evento, tiene un instante de *comienzo* y un instante de *finalización*, y el evento *está transcurriendo* en todos los instantes entre estos dos. Esta consideración, que es evidente para flujo de tiempo discreto, puede ser extendida de forma rigurosa para flujos de tiempo densos o continuos, utilizando las nociones de *presente real, pasado real y futuro real* introducidas en [EdG92]. De esta forma, podemos caracterizar los eventos mediante sus instantes de inicio, finalización, y transcurso. Su formalización se realiza en la extensión de $\mathcal{L}_p$, a la que denotamos $\widehat{\mathcal{L}_p}$, y que definimos a continuación:

Para cada evento atómico $\alpha$ denotamos:

- $\uparrow\alpha$ para: *en este instante comienza el evento* $\alpha$.
- $\downarrow\alpha$ para: *en este instante finaliza el evento* $\alpha$.
- $\overrightarrow{\alpha}$ para: *en este instante el evento* $\alpha$ *está transcurriendo.*

[5] Esto nos permitirá, en LNint de primer orden, hacer aserciones como, por ejemplo "*el presidente de USA en 1962 murió en 1963*", que se representará:

$$Pr(?, USA)\ at\ \underline{1962} \rightarrow M(?)\ at\ \underline{1963}$$

Estas definiciones de *inicio, final* y *transcurso* para los átomos de eventos, se extiende de forma adecuada al lenguaje de eventos.

### 2.3.4. El lenguaje $\mathcal{L}$ de LNint

Para establecer el lenguaje definitivo, $\mathcal{L}$, de LNint, definimos una función de traducción $\mathcal{T}r$ de $\mathcal{L}_{int}$ a $\widehat{\mathcal{L}_p}$ como sigue:

$$\mathcal{T}r : \mathcal{L}_{int} \to \widehat{\mathcal{L}_p}$$

donde $\mathcal{T}r(\mathcal{A}) = \uparrow\mathcal{A} \vee \overrightarrow{\mathcal{A}} \vee \downarrow\mathcal{A}$.

Con esta función tenemos "versiones de puntos" de las fórmulas de $\mathcal{L}_{int}$.

En $\mathcal{L}$ introducimos como definida la conectiva buscada, para dar un tratamiento temporal absoluto a los eventos:

Dada $\mathcal{A} \in \mathcal{L}_{int}$, leemos $\mathcal{A}\, at_I\, [\underline{m}, \underline{n}]$ como *el evento* $\mathcal{A}$ *se da exactamente en el intervalo* [m, n] y se define por

$$\mathcal{A}\, at_I\, [\underline{m}, \underline{n}] = (\uparrow\mathcal{A} \wedge \downarrow\mathcal{A} \approx^+ \underline{n})\ at\ \underline{m}$$

## 2.4. La Semántica Topológica

Como hemos indicado, dotamos a la lógica temporal de una nueva semántica a la que llamamos semática topológica por estar fuertemente basada en la naturaleza *topológica* del tiempo.

### 2.4.1. Semántica of LN

La semántica para LN se define considerando a $(\mathbb{Z}, \leq)$ como flujo del tiempo y utilizando como conceptos claves los siguientes:

Para toda fórmula $A$ y todo instante de tiempo $t$,

$$m^+_{tA} = \text{mín}\{t' > t \text{ tal que } A \text{ es cierto en } t'\}$$

y

$$m^-_{tA} = \text{máx}\{t' < t \text{ tal que } A \text{ is cierto en } t'\}$$

convenimos que $\text{mín}\,\emptyset = +\infty$, que $\text{máx}\,\emptyset = -\infty$

Una *interpretación temporal* para LN es una función que asocia a cada átomo $p$ un subconjunto $h(p)$ de $\mathbb{Z}$:

$$h\colon \Omega \longrightarrow 2^{\mathbb{Z}}$$

La función $h$ puede ser extendida a toda fbf de LN como sigue:

$$\begin{array}{rcl}
h(\top) & = & \mathbb{Z} \\
h(\bot) & = & \emptyset \\
h(\neg A) & = & \mathbb{Z} \setminus h(A) \\
h(A \vee B) & = & h(A) \cup h(B) \\
h(A \wedge B) & = & h(A) \cap h(B) \\
h(A \to B) & = & \big(\mathbb{Z} \setminus h(A)\big) \cup h(B) \\
h(A \prec B) & = & \{t \in \mathbb{Z} \mid m^+_{tA} < m^+_{tB}\} \\
h(A \succ B) & = & \{t \in \mathbb{Z} \mid m^-_{tA} > m^-_{tB}\} \\
h(A \approx^+ B) & = & \{t \in \mathbb{Z} \mid m^+_{tA} < +\infty \text{ y } m^+_{tA} = m^+_{tB}\} \\
h(A \approx^- B) & = & \{t \in \mathbb{Z} \mid m^-_{tA} > -\infty \text{ y } m^-_{tA} = m^-_{tB}\} \\
h(A \preceq B) & = & \{t \in \mathbb{Z} \mid m^+_{tA} < +\infty \text{ y } m^+_{tA} \leq m^+_{tB}\} \\
h(A \succeq B) & = & \{t \in \mathbb{Z} \mid m^-_{tA} > -\infty \text{ y } m^-_{tA} \geq m^-_{tB}\}
\end{array}$$

Como se puede observar, esta semántica permite una lectura formal directa de las conectivas.

Una fórmula $A$ de LN es *válida en una interpretación temporal* $h$, denotada $\models_h A$, si $h(A) = \mathbb{Z}$. Una fórmula $A$ es *válida*, denotada $\models A$, si $\models_h A$ para toda interpretación temporal $h$.

### 2.4.2. Semántica de $\mathbb{R}$LN.

Definimos la *semántica topológica* para $\mathbb{R}$LN considerando $(\mathbb{R}, \leqslant)$ como el flujo del tiempo, donde $\mathbb{R}$ es el conjunto de los números reales.

Para toda fbf A y todo instante de tiempo $t$ en $\mathbb{R}$, se define $i^+_{tA}$ como el inferior de los instantes $t'$ posteriores a $t$ donde $A$ es cierto e $i^-_{tA}$ como el superior de los instantes $t'$ anteriores a $t$ donde $A$ es cierto. Es decir:

$$i^+_{tA} = Inf\{t' \in \mathbb{R} \mid t' \geqslant t \text{ y } A \text{ es cierto en } t'\ \}$$
$$i^-_{tA} = Sup\{t' \in \mathbb{R} \mid t' \leqslant t \text{ y } A \text{ es cierto en } t'\ \}$$

Si $A$ es verdadero en $i^+_{tA}$ e $i^+_{tA} \neq t$, lo denotamos $m^+_{tA}$ (para indicar que es un máximo). Análogamente, si $A$ es verdadero en $i^-_{tA}$ e $i^-_{tA} \neq t$, lo denotamos $m^-_{tA}$ (para indicar que es un mínimo).

Definimos una *interpretación temporal* para $\mathbb{R}$LN como una función que asocia a cada átomo $p$, un subconjunto $h(p)$ de $\mathbb{R}$:

$$h_{\mathbb{R}}\colon \Omega \longrightarrow 2^{\mathbb{R}}$$

La extensión de la función $h_{\mathbb{R}}$ para las conectivas clásicas es idéntica que para la función $h$ de LN. Para las conectivas temporales se define la extensión como sigue:

$$\begin{aligned}
h_{\mathbb{R}}(A \preceq B) &= \{t \in \mathbb{N} \mid m^+_{tA} \leqslant i^+_{tB}\} \\
h_{\mathbb{R}}(A \succeq B) &= \{t \in \mathbb{N} \mid m^-_{tA} \geqslant i^-_{tB}\} \\
h_{\mathbb{R}}(Ac^+(A)) &= \{t \in \mathbb{N} \mid i^+_{tA} = t\} \\
h_{\mathbb{R}}(Ac^-(A)) &= \{t \in \mathbb{N} \mid i^-_{tA} = t\}
\end{aligned}$$

Una fórmula $A$ es *válida en una interpretación temporal* $h_{\mathbb{R}}$, denotada $\models_{h_{\mathbb{R}}} A$, si $h_{\mathbb{R}}(A) = \mathbb{R}$. Una fórmula $A$ es *válida*, denotada $\models A$, si $\models_{h_{\mathbb{R}}} A$ para toda interpretación $h_{\mathbb{R}}$.

#### 2.4.3. Semántica de LNint

Dado que todas las expresiones del lenguaje de LNint están basadas en expresiones de puntos, la semántica topológica de LNint coincide con la de LN. Destacamos la extensión de la función de interpretación a las conectivas de tratamiento absoluto del tiempo:

$$h(A\,at\,\underline{m}) = \begin{cases} T & \text{si } m \in h(A) \\ \emptyset & \text{en otro caso} \end{cases}$$

$$h(\mathcal{A}\,at_I\,[\underline{m},\underline{n}]) = \begin{cases} T & \text{si } [m,n] \in H(\mathcal{A}) \\ \emptyset & \text{en otro caso} \end{cases}$$

donde denotamos con $H(\mathcal{A})$ el conjunto de los intervalos $[t_1, t_2]$ en los que $\mathcal{A}\ at_I\ [\underline{t_1}, \underline{t_2}]$ es verdadera.

Obsérvese como la interpretación de estas conectivas es independiente del instante actual, lo que confirma el carácter absoluto de las conectivas.

## 3. Deducción Automática

Nuestra segunda línea de investigación aborda de deducción automática en Lógica Temporal. En esta búsqueda, nos hemos centrado en extender a la lógica temporal el uso de implicantes e implicados unitarios [6] en Lógica Clásica [GMO$^+$99a, GMO$^+$99b] en el marco de la Metodología de Demostración Automática TAS. [7]

Las propiedades a destacar de los métodos TAS son las siguientes:

— Incorporan un conjunto de *reducciones* [8] de la fórmula en estudio (y de sus subfórmulas), reduciendo la complejidad del problema original.

  Estas reducciones pueden ser incluidas en cualquier método de deducción automática. En particular, nosotros lo hemos aplicado en métodos por refutación tipo Gentzen, desarrollando una familia de demostradores para lógica clásica [AdGO94], lógica multivaluada [AOdGV99] y diversos tipos de lógicas temporales (Lógica Temporal Minimal [EdG95] y Lógica Temporal sobre Tiempo Discreto e Infinito [EdG93b]).

— Las reducciones se aplican tras efectuar test que utilizan la información proporcionada por el conjunto de implicantes e implicados de la fórmula.

  Dicha información se incorpora en cada uno de los nodos del árbol sintáctico de la fórmula analizada, y se almacena en una estructura de tipo lista denominada $\Delta$-lista.

— Las reducciones son de dos tipos:

  — Reducciones que conservan el significado de la fórmula analizada, $A$, y la transforman en una constante lógica ($\top$ ó $\perp$) o un literal, equivalente a $A$.

[6] Fijada un lógica $\mathcal{L}$ y una fórmula $A$ $\mathcal{L}$, se dice que un literal $\ell$ es un implicante si y sólo si $\models \ell \to A$ y decimos que $\ell$ es un implicado si y sólo si $\models A \to \ell$.

[7] Denominada así porque los métodos en ella desarrollada, se describen en términos de *Transformaciones de Árboles Sintácticos*.

[8] Estas reducciones tienen coste a lo sumo polinómico.

— Reducciones que conservan la satisfacibilidad de la fórmula analizada, $A$, y la transforman en una fórmula de menor tamaño $B$ simultáneamente satisfacible con $A$.

— Son métodos de construcción de modelos. [9] Esta característica, adquiere especial relevancia, no sólo por su interés para las aplicaciones, sino también porque la construcción del modelo se genera de modo natural durante la ejecución sin más que guardar la información aportada por los implicados e implicantes utilizada en las transformaciones que conservan la satisfacibilidad.

La extensión del uso de implicantes e implicados de la lógica clásica a la lógica temporal, requiere un extenso estudio teórico hasta llegar a disponer de $\Delta$-listas con las propiedades deseadas. En [dGEC99a, dGEC99b] incluimos dicho estudio, realizado con dos objetivos:

— Caracterizar la forma que han de tener las $\Delta$-listas para almacenar la máxima información sobre implicantes e impliciados con la mínima cantidad de elementos.

— Definir un conjunto de operadores para la construcción y el uso eficiente de las $\Delta$-listas .

En [dGEC99b] se incluye el estudio para el fragmento de futuro de la lógica temporal de tiempo discreto lineal e infinito. En [dGEC99a] se extienden los resultados a la lógica con conectivas de pasado y futuro, demostrando que, a pesar de la complejidad teórica, se mantiene la eficiencia de los operadores que manipulan las $\Delta$-listas. La extensión a la lógica LN es nuestro trabajo actual.

### 3.1. Implementación del Demostrador

Con el objetivo de poder analizar la viabilidad del uso de nuestras propuestas en aplicaciones, hemos la implementación de nuestros demostradores usando el lenguaje C++, con resultados satisfactorios [dGEM96], [dGEM97]. Recientemente, hemos desarrollado versiones de los demostradores TAS temporales usando el sistema MAUDE y CAMEL para la lágica clásica proposicional.

[9] Es decir, proporcionan un modelo de la fórmula, si esta es satisfacible.

### 3.2. Aplicaciones

Recientemente hemos establecido un convenio de colaboración con la empresa Espaceland en Málaga que recoge, de forma muy completa, gran parte de las aplicaciones que nuestros resultados pueden cubrir.

Espaceland está desarrollando un sistema de información que permite la gestión de documentos digitalizados. [10] Nuestros resultados permiten añadir a la base de datos la capacidad de gestión temporal de la información, con las siguientes características:

— Se usa lógica temporal (LN o LNint según el módulo) para la especificación de las restricciones temporales que debe satisfacer el sistema.

— Disponemos de un lenguaje de consulta a bases de datos relacionales temporal, extensión del lenguaje SQL usando las conectivas de LNint.

— Nuestro objetivo es utilizar los métodos TAS temporales para dotar al sistema de un optimizador de consultas temporales que, de manera análoga a cómo funciona un optimizador de consultas clásico, permita tratar la información temporal contenida en la consulta, mejorando el tiempo de acceso a la base de datos. [11]

## Referencias

[AdGO94] Gabriel Aguilera, Inma P. de Guzmán, and Manuel Ojeda. Automated model building via syntactic trees transformations. In *Proceedings of the CADE-12*

[10] El objetivo es aligerar a las empresas y organizaciones del coste añadido de disponer de un espacio para el almacen de documentos, presentando además un sistema de información que agilice la gestión de dichos documentos.

[11] Esta técnica está conectada con los trabajos de nuestro grupo en transformación algebráica de programas.

*Workshop on Automated Model Building*, pages 4–10, Nancy (France), June 1994.

[All83] J. F. Allen. Maintaining knowledge about temporal intervals. *Communications of the ACM*, pages 832–843, 1983.

[AM87] M. Abadi and Z. Manna. Temporal logic programming. In *Symposium of Logic Programming*. IEEE, 1987.

[AOdGV99] G. Aguilera, M. Ojeda, I. P. de Guzmán, and A. Valverde. Reducing signed propositional formulas. *Soft Computing*, 2(4), 1999.

[BG92] A. Burrieza and I. P.de Guzmán. A new algebraic semantic approach and some adequate connectives for computation with temporal logic over discrete time. *Journal of Applied non Classical Logic*, 2, 1992.

[BPK86] H. Barringer, A. Pnuelli, and R. Kuiper. A really abstract concurrent and its temporal logic. In *POPL*. ACM, 1986.

[dGEC99a] Inma P. de Guzmán, Manuel Enciso, and Pablo Cordero. Structure theorems for closed sets of implicates/implicants in temporal logic. *LNAI. EPIA'99*, Spinger-Verlag, 1999.

[dGEC99b] Inma P. de Guzmán, Manuel Enciso, and Pablo Cordero. A temporal negative normal form which preserves implicants and implicates. *Journal of Applied non Classical Logics*, To appear, 1999.

[dGEM96] Inma P. de Guzmán, Manuel Enciso, and Juan F. Moncada. Executing intensional logic with the tas tool. In *Int. Joint Conf. on Declarative Programming Prode'96*, 1996.

[dGEM97] Inma P. de Guzmán, Manuel Enciso, and Juan F. Moncada. Demostración automática de teoremas en lógica modal temporal: una nueva aproximación al método de gentzen. In *Escuela de Verano de Informática*, 1997.

[dGR95] I. P. de Guzmán and C. Rossi. Lnint: a temporal logic that combines points and intervals and the absolute and relative approaches. *Journal of the IGPL*, 3(5), 1995.

[EdG92] M. Enciso and I. P. de Guzmán. Controlando el tiempo continuo. In Manuel Hermenegildo and Juan J. Moreno, editors, *Primer Congreso Nacional de Programación Declarativa, PRODE'92*, 1992.

[EdG93a] M. Enciso and I. P. de Guzmán. An application of temporal logic over continuous time to databases. control of interferences due to the concurrent execution of transactions following the timestamping method. In *TARRAT*, Blanes (Gerona), Septiembre 1993.

[EdG93b] M. Enciso and I. P. de Guzmán. A new and complete temporal. *Lect. Notes in Artif. Intelligence no. 838*, pages 198–216, 1993.

[EdG95] M. Enciso and I. P. de Guzmán. A new and complete automated theorem prover for temporal logic. In *IJCAI-Workshop on executable temporal logics*, 1995.

[Gab89] D. M. Gabbay. *The declarative past and imperative future*, volume 398 of *Lecture Notes in Computer Science*. Springer Verlag, 1989.

[Gab92] D. M. Gabbay. *Temporal Logic: Mathematical Foundations*. Oxford University Press, 1992.

[GMO+99a] G. Gutiérrez, J. Martinez, M. Ojeda, I. Pérez de Guzmán, and A. Valverde.

Reduction theorems for boolean formulas using $\delta$-trees. Enviado a publicación, 1999.

[GMO$^+$99b] G. Gutiérrez, J. Martinez, M. Ojeda, I. Pérez de Guzmán, and A. Valverde. Reductions for non-clausal theorem proving. Por aparecer en Theoretical Computer Science, 1999.

[Sho88] Y. Shoham. *Reasoning about Change. Time and Causation from the Standpoint of Artificial Intelligence.* MIT Press, 1988.

[Wol93] P. Wolper. Esprit basic research action react(6021): Building correct reactive systems. *EATCS Bulletin*, 50, 1993.